
# 微架构设计之微博计数器服务

杜传赢 @cydu
chuanying@staff.sina.com.cn
http://weibo.com/cydu
http://blog.cydu.net/


- 新浪微博 -- 中国最具影响力的微博产品
  - 信息实时聚合平台
    - 数千个消息源的聚合 & 千万级的消息接收者
  - 巨大的数据量
    - 微博数据量: 千亿级
    - 高峰微博增加量: 每秒3万条(春节)
- 巨大的访问量
  - 每天动态API请求量超过300亿次
  - 每秒数百万次的服务调用(包括内部调用)

# 微博计数器服务


我的首页 新浪微博-随时随地分享身边的新鲜事儿
新浪微博
首页
应用
微吧
游戏
搜索微博、找人
cydu
未读计数
2条新评论，查看评论
2位新粉丝，查看粉丝
3条新@我，查看@我
首页
提到我的
评论
表态
私信
收藏
发给我的
分组
密友
weiboplat
weibotech
weibo
sina
展开
最近使用
微数据
Qing
有什么新鲜事想告诉大家？
发言请遵守社区公约，还可以输入111字
微架构设计之微博计数器服务 by @cydu http://blog.cydu.net/2012/09/weibo-counter-service-design-2.html
用户计数
801 关注
2287 粉丝
1777 微博
公开
发布
回顾你的微博点滴 HOT
微博
动态
查找作者、内容或标签
时间排序
全部
原创
图片
视频
音乐
共搜索到 1 条微博
«返回查看全部
可能感兴趣的人
找人
炳叔
6个间接关注
+关注
包括徽剑、cnsns、虎嗅网等
南非蜘蛛
8个共同好友
经纬张颖
6个间接关注
鞍钢郭明义
全国优秀共产
更多»
[活动]晒彩票赢奖品
WeiT
微博平台架构
#微架构设计# 大家还记得上周的微博计数器设计的小练习么？ 经过一周的思考，大家快来看看@cydu 同学 的思路吧，1000亿的数字，每秒100W次查询，单机能搞得定吗？ http://t.cn/zWDtOnE
Weibo Counter
9月7日15:05 来自专业版微博 | 举报
(1) | 转发(251) | 收藏 | 评论(45)
微博计数
私信聊天[134]

第一版

从无到有

# 架构挑战：开发速度

**2009.7月** 正式立项

**2009.8.28** 新浪微博 上线 t.sina.com.cn


SELECT count(**mid**) FROM Status WHERE **uid** = 888

- **mid** => 每条微博的唯一标识(64bit)
- **uid** => 每位用户的唯一标识(64bit)

# 实时计算


SELECT count(**mid**) FROM Status WHERE **uid** = 888

- **mid** => 每条微博的唯一标识(64bit)
- **uid** => 每位用户的唯一标识(64bit)

# 计数索引

郑渊洁V：我今晚7点30分作为评审团成员参加湖南卫视快乐男声总决赛直播。我现在想了解李炜、刘心和武艺这三位选手谁的支持者数量多。支持刘心的请转发此贴，我按本贴转发量计算刘心的粉丝数量。

2010-9-10 13:10 来自新浪微博 转发(2257106) | 收藏 | 评论(9877)

SELECT repost_count, comment_count FROM Status_count WHERE **mid** = 888

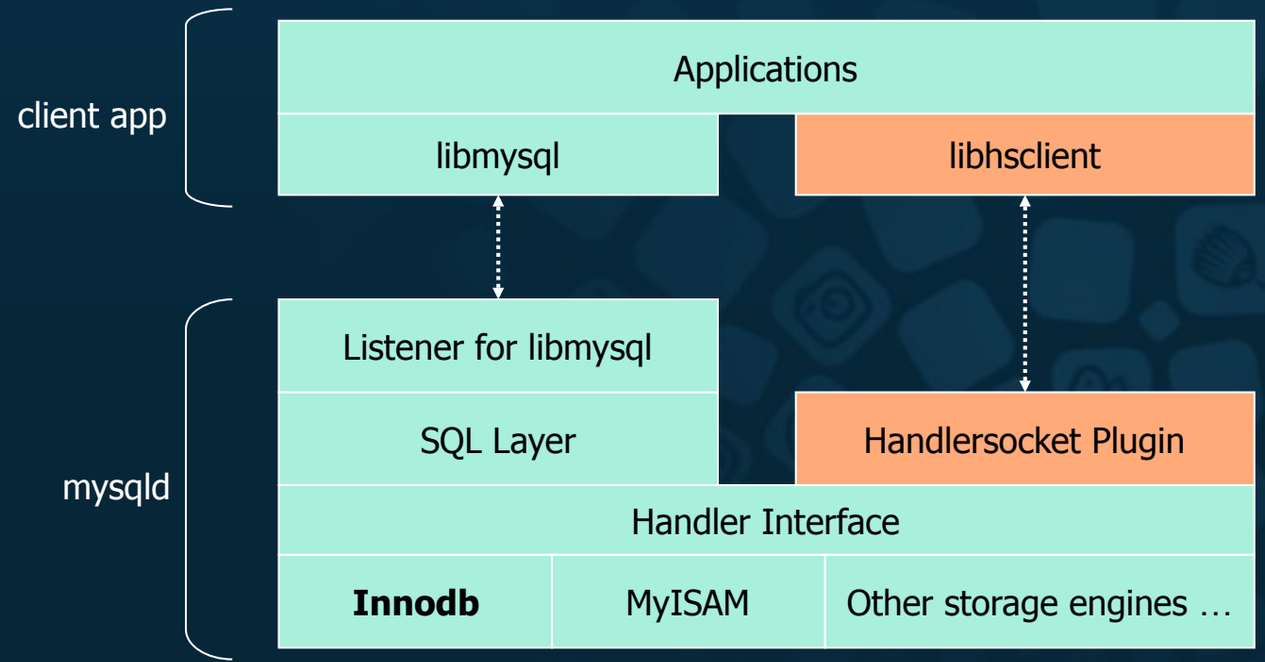
- 微博变多：实时**Count**代价高

2010-9-10 转发 2257106 | 收藏 | 评论(9877)

- 流量增大：**Feed**页出微博计数
- 数据一致性？ 如何更新？

# Scale UP

**硬件加速**





# MySQL HandleSocket

FUSION-IO

- 摩尔定律: 每隔18个月, 性能提升一倍!
- Jim Gray:
  - Tape is dead, disk is tape, **flash** is disk, **ram** locality is **king**

# 小结：切忌过度优化

**先建设，再优化**

数据量(1000亿)

开发速度

MySQL

FUSION-IO

访问量(100W rps)

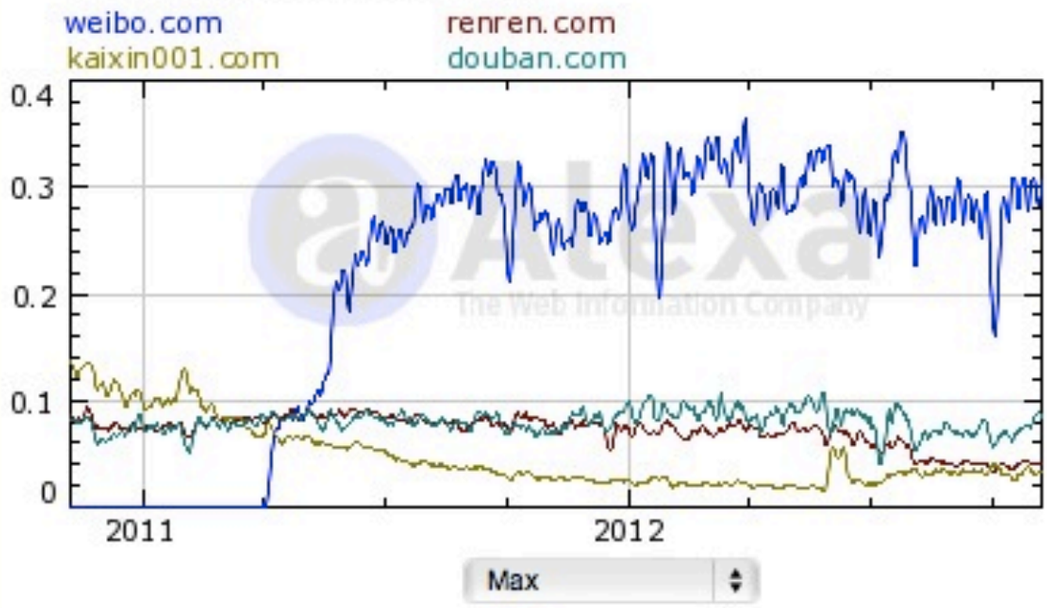
第二版(I)

从小到大

# 架构挑战1：访问量

Traffic Rank | Reach % | **Pageviews %** | Pageviews/User | Bounce % | Time on Site | Search %

Daily Pageviews (percent)




Max

Estimated percentage of global pageviews on weibo.com:

| | Pageviews % | Change |
|---|---|---|
| **Yesterday** | 0.3166% | -2.85% ↓ |
| **7 day** | 0.2987% | +1.9% ↑ |
| **1 month** | 0.2953% | +17.69% ↑ |
| **3 month** | 0.27882% | -11.34% ↓ |

Compare **weibo.com** to:

renren.com | kaixin001.com | douban.com | Compare

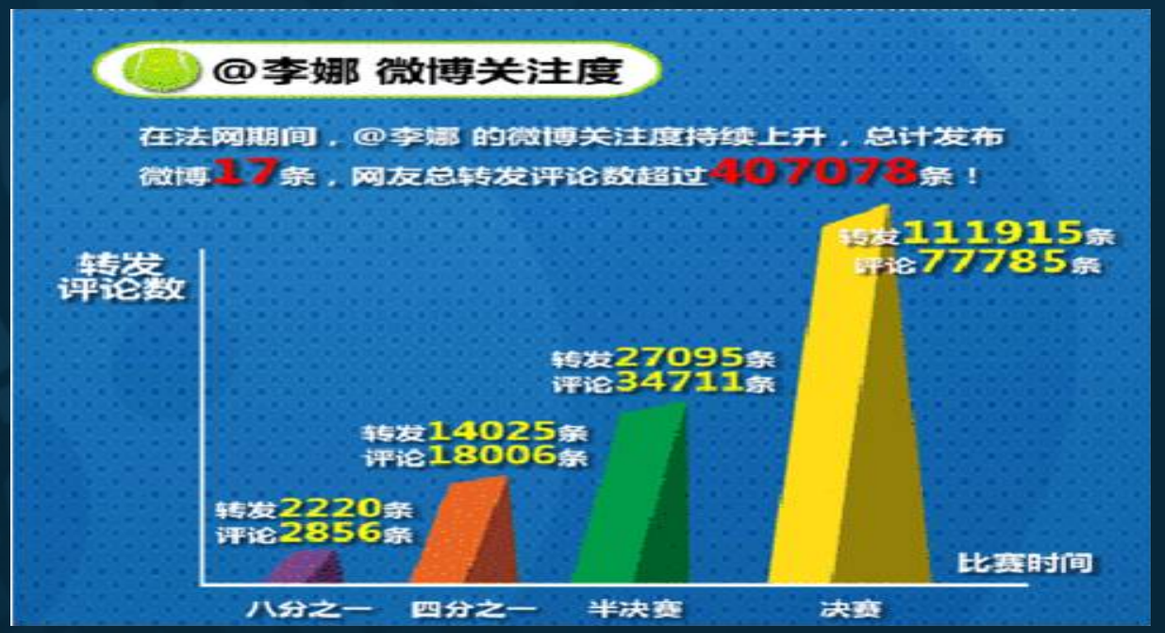
Learn more about Alexa Traffic Stats.

Share this: facebook Tweet +1

# weibo.com独立访问量已是国内第 六 大网站

# 突发热点更新


“伤”别 刘翔
刘翔，一个时代的传奇
截至8月7日21点
新浪微博刘翔话题讨论量
2724万
截至8月7日21点
刘翔相关视频总播放量
2890万

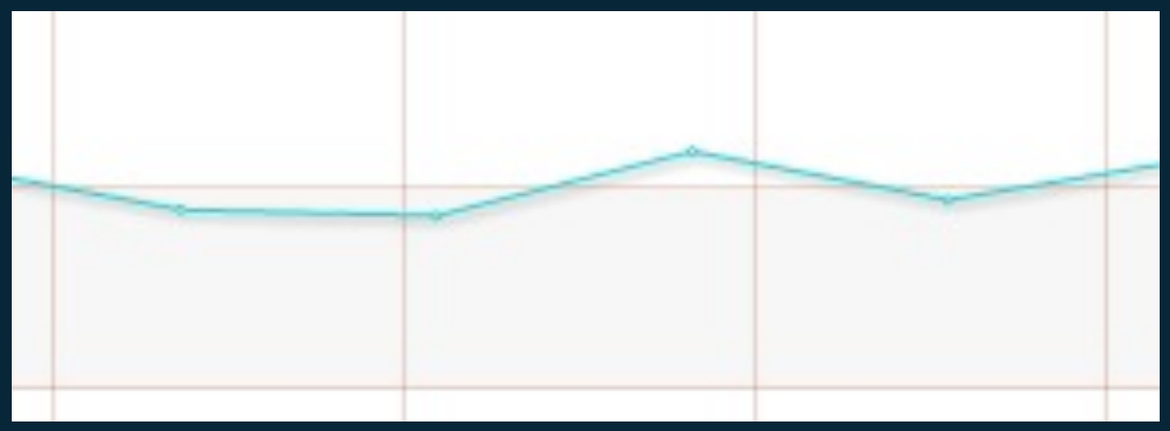

@李娜 微博关注度
在法网期间，@李娜 的微博关注度持续上升，总计发布微博17条，网友总转发评论数超过407078条！
转发评论数
转发2220条 评论2856条
转发14025条 评论18006条
转发27095条 评论34711条
转发111915条 评论77785条
比赛时间
八分之一
四分之一
半决赛
决赛


龙年快乐

# 异步更新

## ID分配器 + 消息队列

**MessageQueue**



- 更新消息队列
  - 消息更新异步化
  - 削峰填谷
  - 避免高峰时数据丢失

# 批量更新

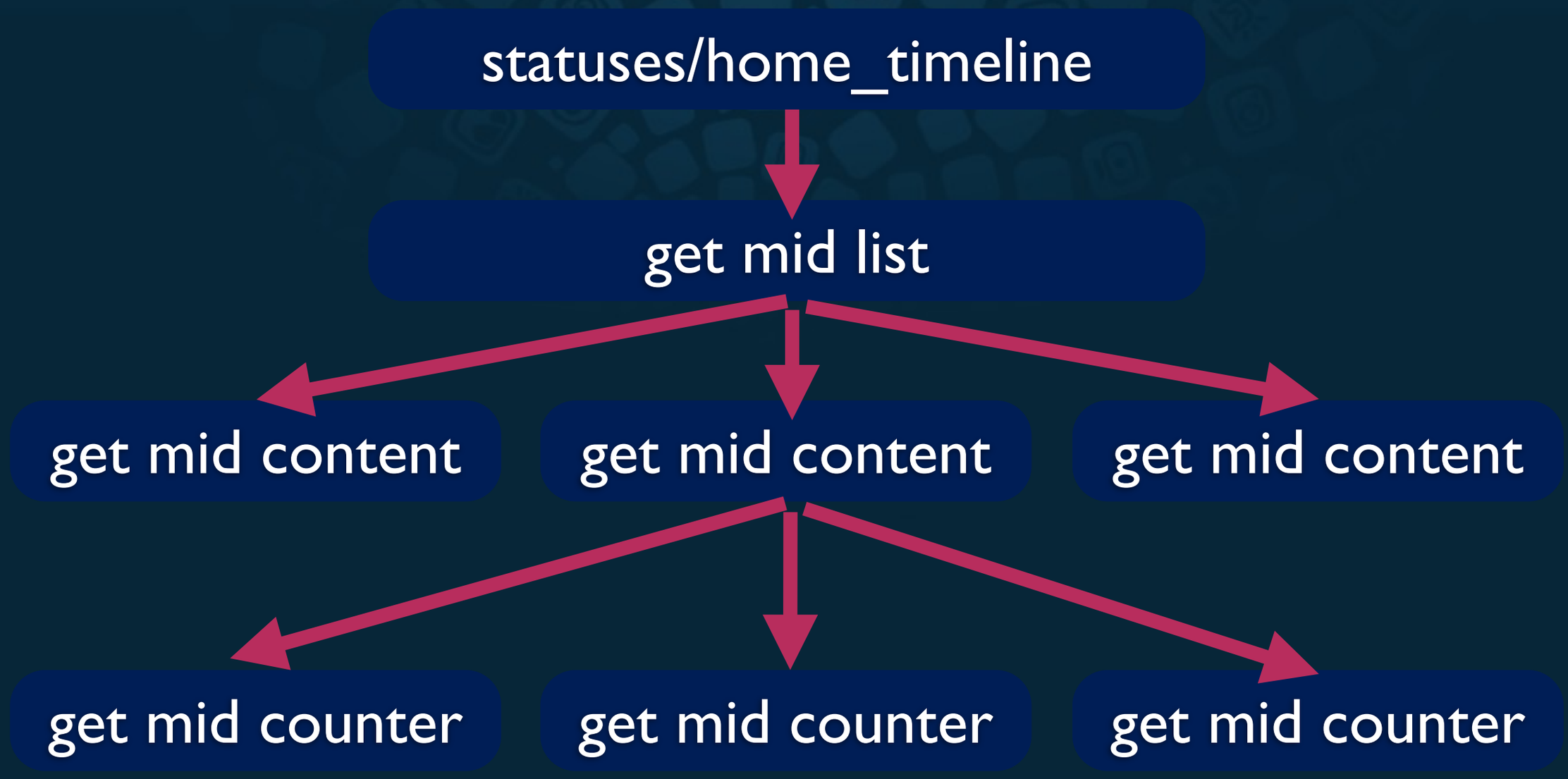
```
UPDATE Status_count SET repost_count =
    repost_count +1 WHERE mid = 888;
UPDATE Status_count SET repost_count =
    repost_count +1 WHERE mid = 888;
UPDATE Status_count SET repost_count =
    repost_count +1 WHERE mid = 888;
```

```
Update Status_count SET repost_count =
repost_count +3 WHERE mid = 888;
```

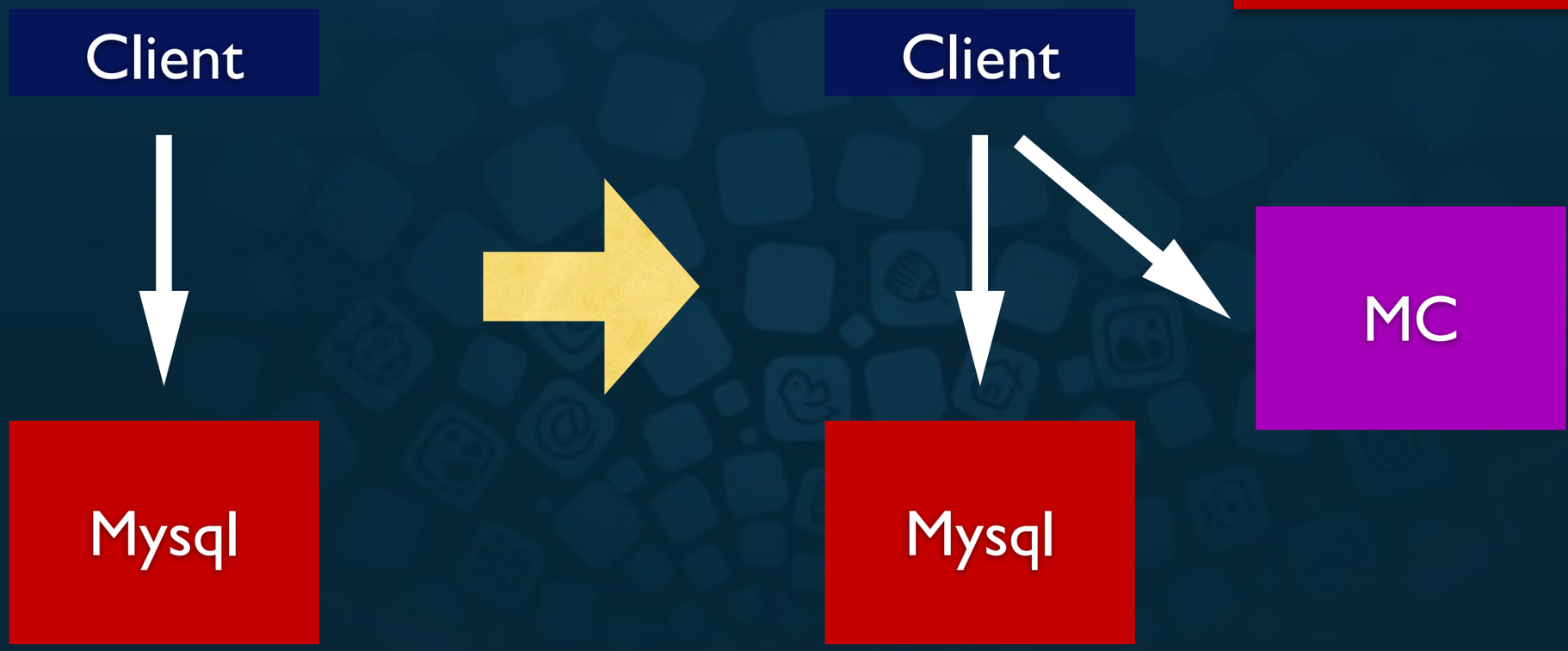
- 批量更新:
  - 取多条消息, **Merge**后再更新, 降低写压力

# 大量读挑战

## 超过15倍的放大效应!

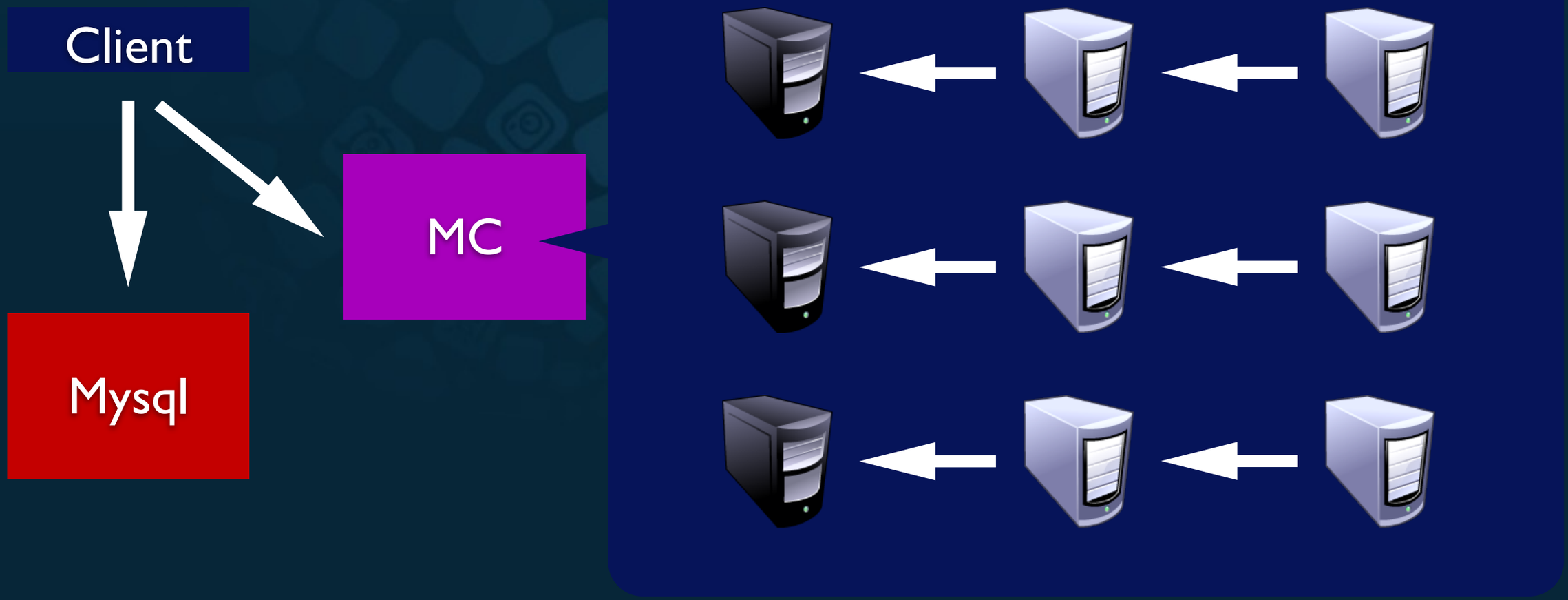


# 缓存优化




- 读写比 && 命中率
  - Cache效率的关键
  - “Cache就像万金油，哪痒你就抹哪，但是千万记得脱了皮鞋再抹！” -- by 朱聪

# MultiGet




- MultiGet Hole ？ More Machines != More Capacity
- Evections (缓存踢出)

第二版(2)

从小到大

**架构挑战1： 访问量**

从少到多

**架构挑战2： 数据量**

新浪微博
weibo.com

用户数（单位：千）

# 水平扩展

Partition by primary key
用户类数据

Partition 1

| uid | fans_num |
|---|---|
| 20 | ... |
| 22 | ... |
| 24 | ... |

Partition 2

| uid | fans_num |
|---|---|
| 21 | ... |
| 23 | ... |
| 25 | ... |

Partition by time/mid
微博类数据

Partition 2

| mid | repost_num |
|---|---|
| 24 | ... |
| 23 | ... |

Partition 1

| mid | repost_num |
|---|---|
| 22 | ... |
| 21 | ... |

# 两层划分

Partition 4

| mid | repost_num |
|---|---|
| 34 | ... |
| 32 | ... |

Partition 3

| mid | repost_num |
|---|---|
| 33 | ... |
| 31 | ... |

Partition 2

| mid | repost_num |
|---|---|
| 24 | ... |
| 22 | ... |

Partition 1

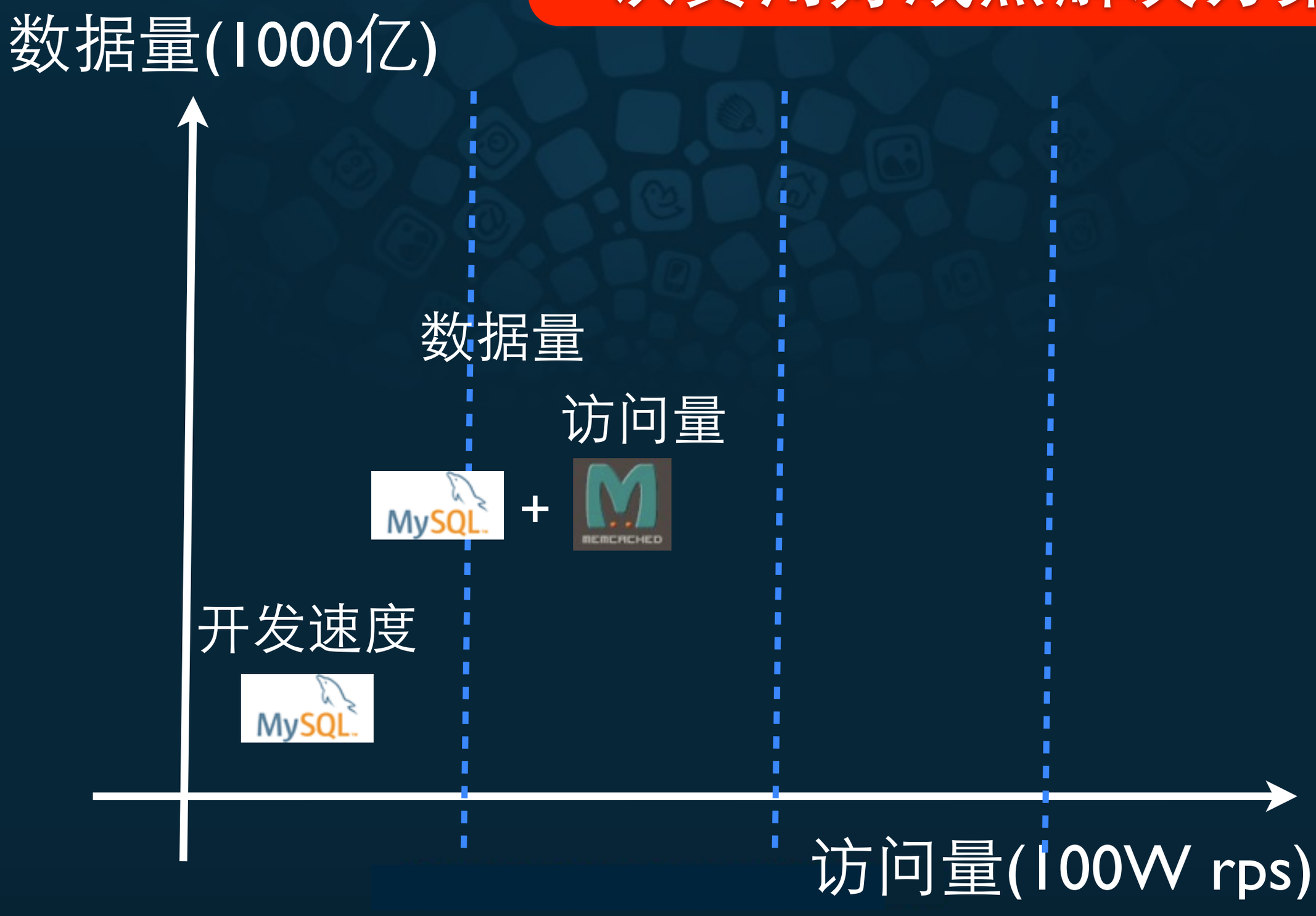
| mid | repost_num |
|---|---|
| 23 | ... |
| 21 | ... |

Partition by mid + time

微博类数据

# 小结：见招拆招，快速迭代

**认真用好成熟解决方案**



# 第二版的问题: Pain Driven Development

- 数据量越来越大, 分表越来越频繁/复杂
  - 操作更复杂, 风险大, 成本也越来越高
- 访问量越来越大, Cache命中率如何提升?
  - 内存使用效率低下(Eg: 字符串,0数据,旧数据)
  - 副本过多时, 数据更新代价高
  - 完全依赖内存, 存储成本高
- 高可用的要求
  - 机器故障时的快速恢复能力及最低的线上影响
- 越来越复杂的需求带来的业务挑战

# 第三版(I)

由粗到细

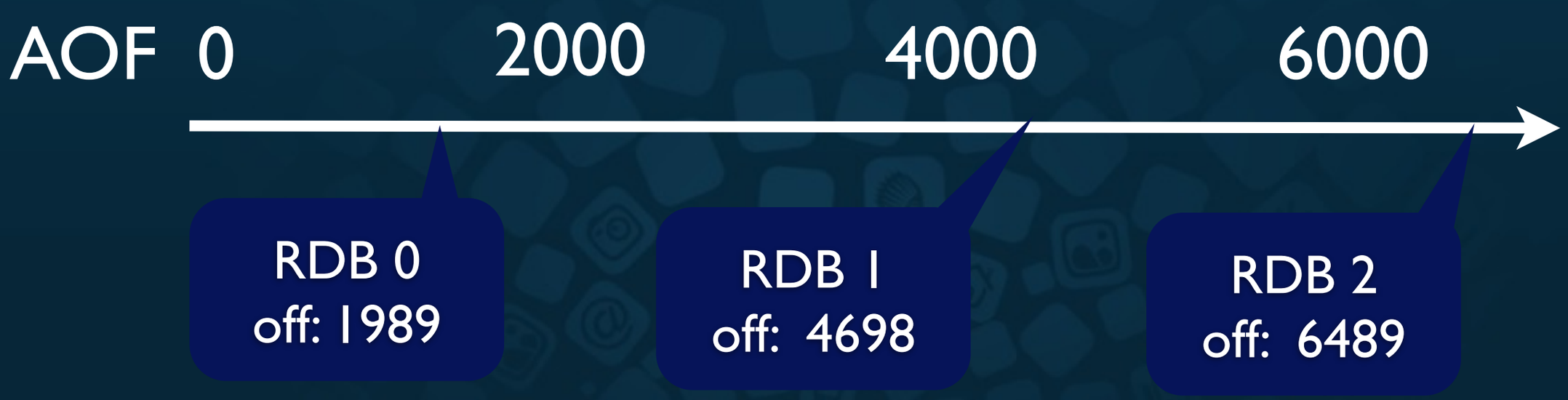
**架构挑战1：高可用**


# REmote DIctionary Server

by @antirez

- 成熟 : 社区活跃 + 生产环境使用多
- 高速 : 全内存, 性能优异 + 数据镜像,快速恢复
- 简单 : 友好的DSL接口 + 丰富的数据结构
- 可控 : 架构简洁 + 代码量不大

# RDB+AOF




- RDB + AOF 同步模式
  - 定期 bgsave 生成RDB
    - 避免AOF Load过慢
  - syncfrom aof offset
    - 避免从头Sync
- 热升级


@果爸果爸 / @Jokea

# 多机房

**IDC 0:**

Web Service (Update)

Message Queue

Mysql Cluster

Transformer

Redis

Web Service (Query)

Web Service (Update)

Message Queue

Mysql Cluster

Transformer

Redis

Web Service (Query)

We

**IDC : 2**

Message Queue

Mysql Cluster

Transformer

Redis

Web Service (Query)

# 灾难降级

A. 正常服务

B. 降级写服务

- 写延迟(堵队列)
- 拒绝写(事后修数据)

C. 降级读服务

- 只读新数据(老数据出0)
- 新数据部分可
- 全部出默认值

D. 服务读写均不可用

- 不出计数

# 数据一致

- 最终一致性!
  - Randomkey check
  - RDB CRC check
  - Digest check (write crc)
  - 定期全量sync
  - 数据基准修复

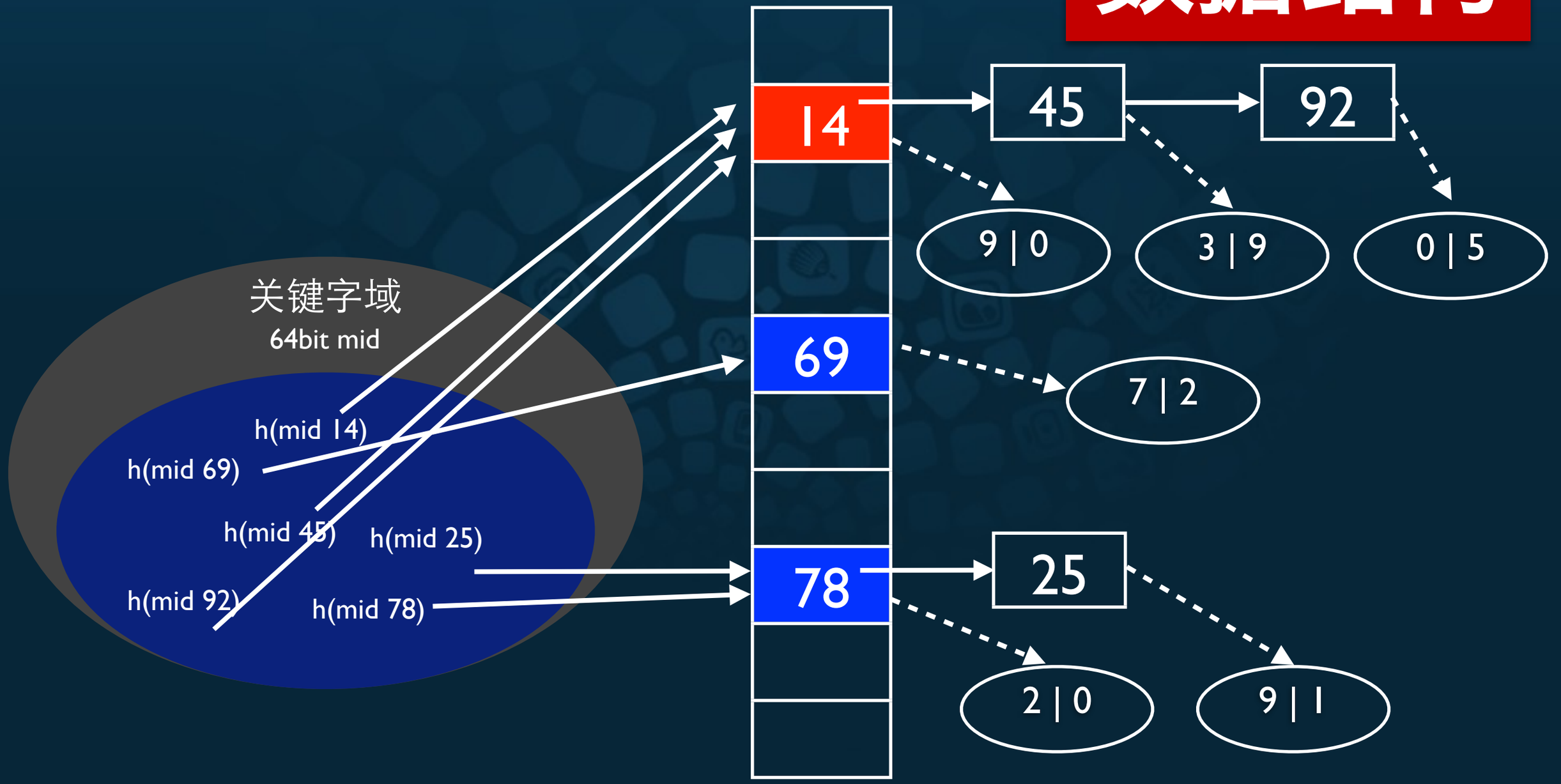
# 第三版 (2)

由粗到细

**架构挑战1：高可用**

**架构挑战2：低成本**

# 数据结构



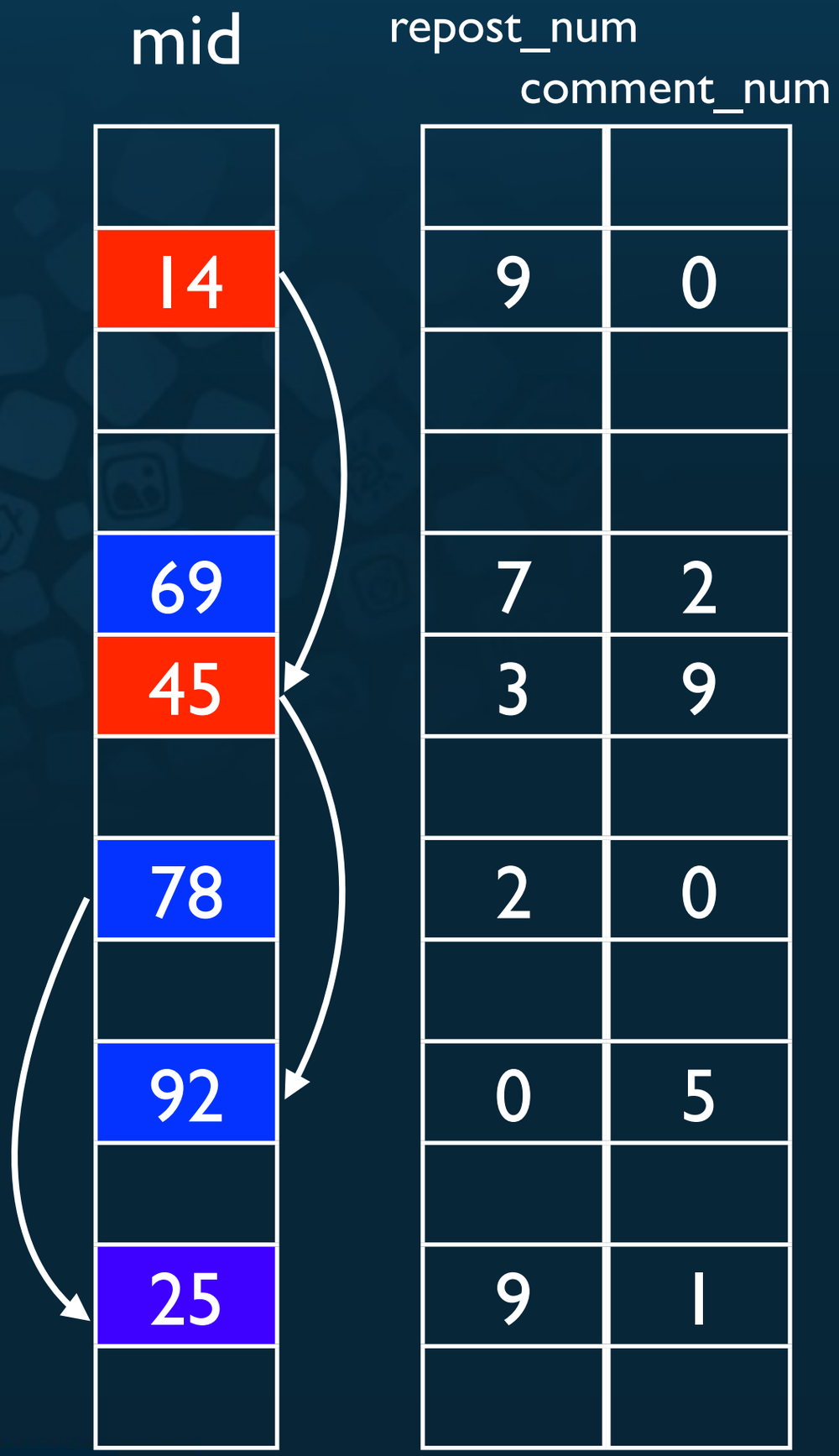

传统的Hash表大量的指针开销

# 开放寻址Hash节省内存

**h(k,i) = (h1(k)+i*h2(k)) % m**

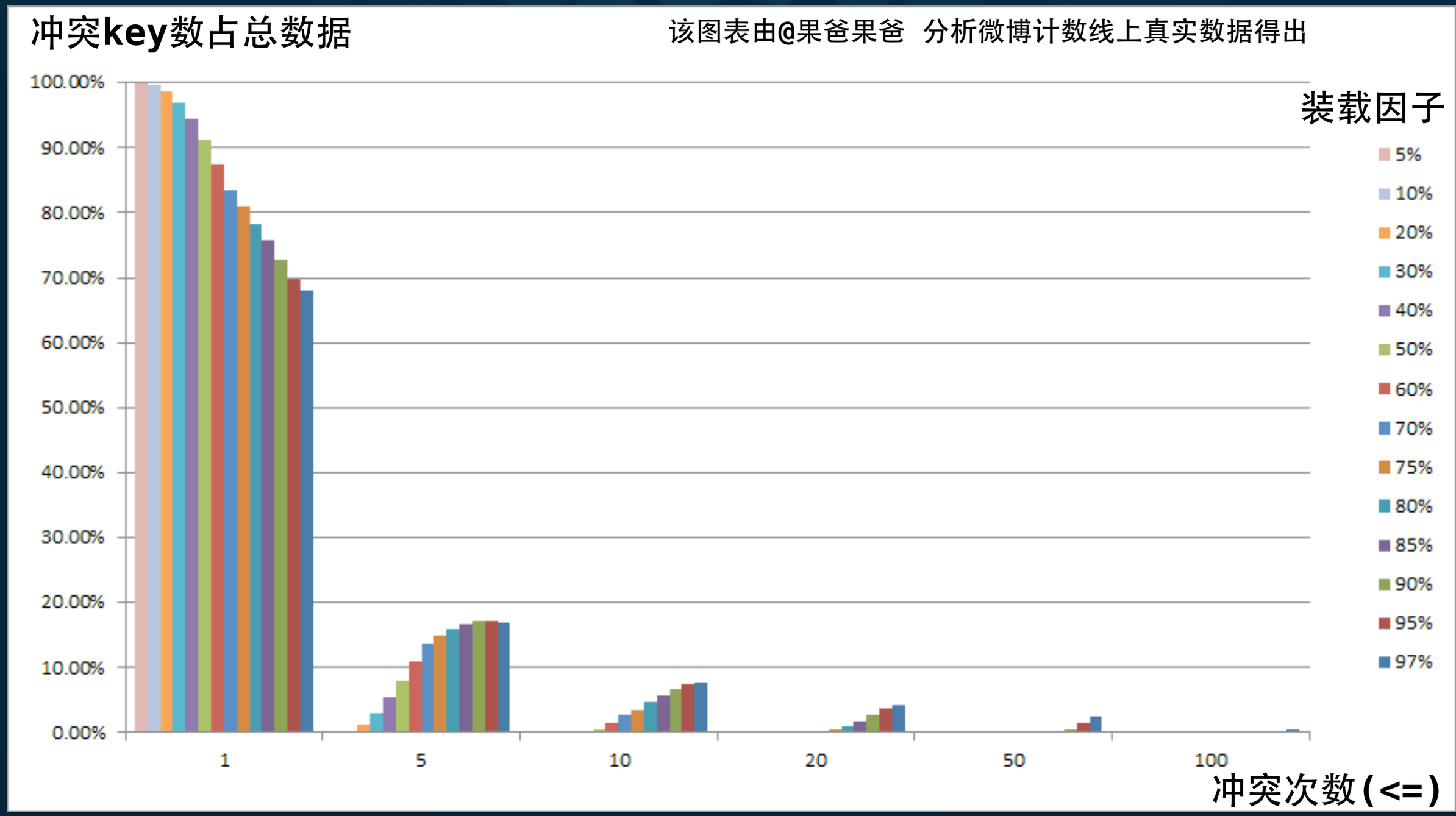
```
struct item {
    int64_t mid;
    u_short repost_num;
    u_short comment_num;
};
```

- Value的长度短于指针长度
- 开放寻址Hash表(双重散列)
- 以节省指针存储



**E[X]：** 期望探查次数　　**α：** 装载因子

$$E[X] = \sum_{i=1}^{\infty} \Pr\{X \geqslant i\} \leqslant \sum_{i=1}^{\infty} \alpha^{i-1} = \sum_{i=0}^{\infty} \alpha^{i} = \frac{1}{1-\alpha}$$



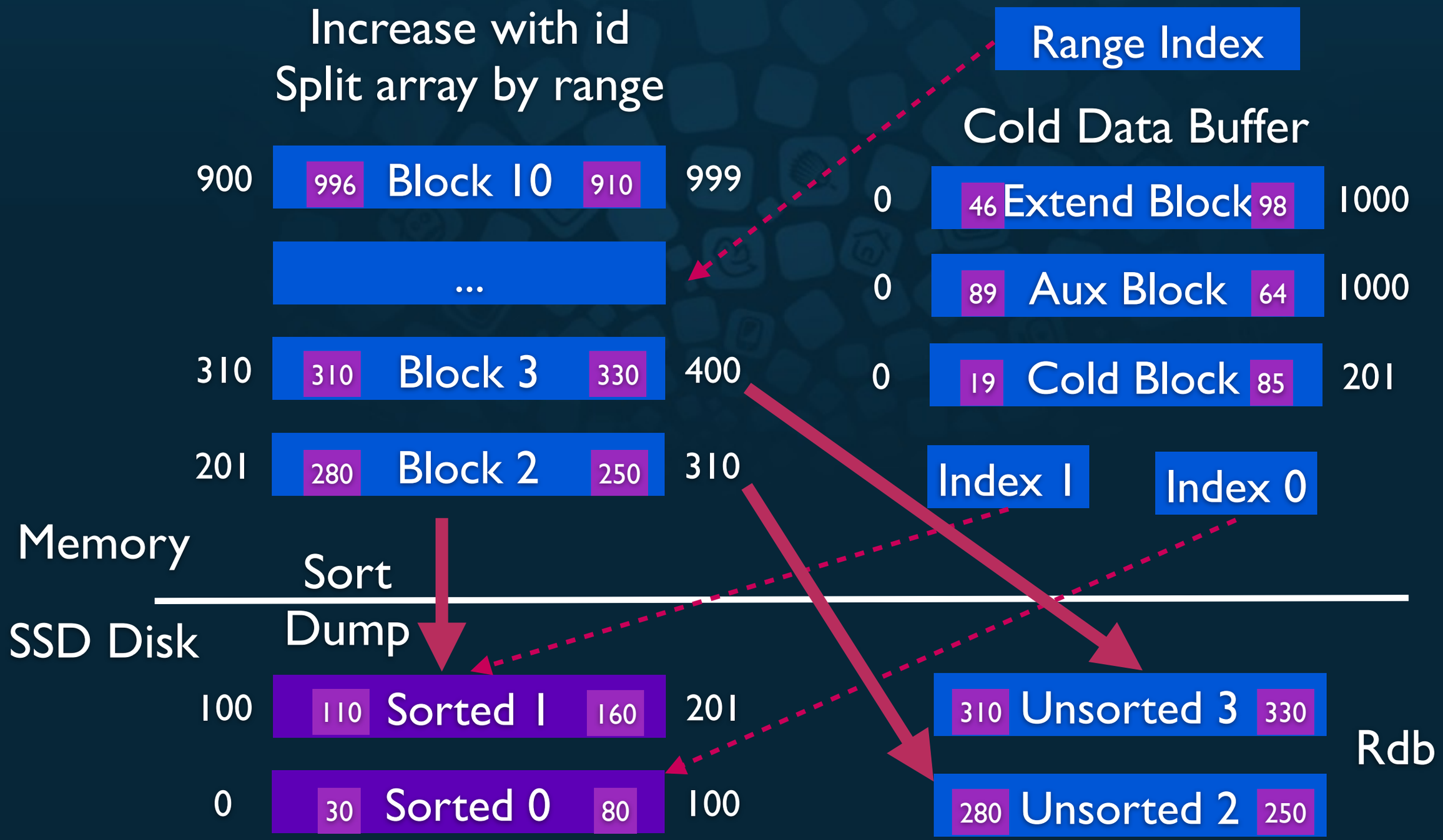
# 数据压缩

- repost_num & comment_num
  - 32bit *2 => 16 bit * 2
- mid
  - string => int64_t
- Value block compress by @吴廷彬
  - Value定长块实时压缩和解压(对上层透明)
- Key prefix compress by @吴廷彬 @drdrxp
  - key基本有序后, 64 bit的key相同前缀(Eg: 高32位)提取

# 还有几个小问题

- repost_num 和 comment_num 只用16位存储,超过65535的转发数和评论数怎么办?
- 采用开放寻址仍会冲突，在极端情况下，冲突加大影响性能怎么办？会有潜在的”死循环”吗?
- 内存不够怎么办？（尽管已经精简使用,但数据量仍很夸张）
- 选取哪些Key放到内存？为什么不用LRU?
- 用户突然大量访问老的微博怎么办?
  - Eg：“转发我的第一条微博”，“去年今日的我”

# Weibo Counter Service新架构



# 第三版(3)

由粗到细

**架构挑战1：高可用**

**架构挑战2：低成本**

**架构挑战3：多变需求**

- 微博计数
  - 评论数 / 转发数
  - 表态数
    - 喜欢数 / 开心数 / 吃惊数 / 悲伤数 / 愤怒数
- 用户计数
  - 关注数 / 粉丝数 / 好友数 / 微博数 / 原创微博数 ...
- 其他计数
  - 未读数 / 提醒数
  - 链接点击数 / 收藏数
  - 会员数 / 应用计数 / 管理类计数...

# 服务化支持

- add counter weibo
- add column weibo mid hint=64 max=64 primarykey
- add column weibo comment hint=16 max=32 default=0 suffix=cntcm
- add column weibo repost hint=16 max=32 suffix=cntrn
- add column weibo attitude hint=8 max=32 suffix=cntan
- set 19089006004.cntcm 987654
- incr 888888.cntrn
- get 123456.cntan
- del 19089006004

# 统计支持

- 每个计数的统计
  - 容量 / 目前使用量
  - getCount / setCount / missCount / hitCount
  - errorCount / fullCount / collisionCount
- 计数中每个列的统计
- 计数中每个Table的统计
- 慢查询的统计

对业务需求，业务状态，服务状态，架构缺陷等更好的理解才能支持更好的决策！

# 小结: 量体裁衣, 精益求精!

# 小结: 量体裁衣, 精益求精

架构没有最好, 只有合适和更优

数据量(1000亿)

开发速度

MySQL

数据量

MySQL + MEMCACHED

访问量

高可用

服务化

低成本

Counter
www.weibo.com

redis

访问量(100W rps)


# Q&A

- 感谢各位, 欢迎各种意见和建议, 当然也包括拍砖!
- 本文中提及计数服务相关设计和实现主要贡献者:
  - @微博平台架构 @果爸果爸 @LinuxQueue @cydu
- 之前的设计总结和讨论见Blog:
  - http://blog.cydu.net/2012/09/weibo-counter-service-design-2.html